On Detection and Visualization Techniques for Cyber Security Situation Awareness

Wei Yu[†], Sixiao Wei[†], Dan Shen[‡], Misty Blowers[*], Erik P. Blasch[*], Khanh D. Pham[II], Genshe Chen[‡], Hanlin Zhang[†], Chao Lu[†]

[†]Computer and Information Sciences Dept., Towson University, Towson MD, 21252
[‡] Intelligent Fusion Technology, Inc, Germantown, MD 20876
[*]Air Force Research Laboratory, Information Directorate, Rome, NY 13441
[II]Air Force Research Laboratory, Space Vehicles Directorate, Kirtland AFB, NM 87117

ABSTRACT

Networking technologies are exponentially increasing to meet worldwide communication requirements. The rapid growth of network technologies and perversity of communications pose serious security issues. In this paper, we aim to developing an integrated network defense system with situation awareness capabilities to present the useful information for human analysts. In particular, we implement a prototypical system that includes both the distributed passive and active network sensors and traffic visualization features, such as 1D, 2D and 3D based network traffic displays. To effectively detect attacks, we also implement algorithms to transform real-world data of IP addresses into images and study the pattern of attacks and use both the discrete wavelet transform (DWT) based scheme and the statistical based scheme to detect attacks. Through an extensive simulation study, our data validate the effectiveness of our implemented defense system.

1. INTRODUCTION

Currently, cyber attacks are growing in frequency, impact, and complexity, raising serious threats to cyber critical infrastructure. In particular, an adversary may hack into enterprise's servers or information systems from communication networks, which can create significant impact by using plentiful attacking techniques, including the distributed denial-of-service (DDoS),[1] botnet,[2–4] and others. Hence, there is an urgent need for cyber security.[5,6] To address issues in cyber awareness,[7–10] we shall develop network security situation awareness techniques to defend against cyber attacks through distributed collaborative monitoring, detection, and mitigation. Such a defense system can characterize, track, and mitigate threats in a timely manner over a network.[11] Note that the situation awareness is generally described as "knowing what is going on around the system and within that knowledge of surroundings and being able to identify which events in those surroundings are important".[12,13] Situation awareness is based on a human mental model, whereas situation assessment is the machine analysis of the scenario.[11,14,15]

A number of research efforts have been made to investigate cyber attacks and defensive techniques.[16–21] For example, Wagner *et al.* in[16] investigated several host-based anomaly detection schemes and introduced a mimicry attack, which allows a sophisticated adversary to avoid the detection by Intrusion Detection Systems (IDS). They also studied a theoretical framework for evaluating the performance of IDS against mimicry attacks. Guoyin *et al.* in[17] proposed an intrusion detection scheme using genetic algorithms. Liang *et al.* in[18] presented an architecture of intrusion detection system and its effectiveness in detecting intruders. Krishnan *et al.* in[19] proposed a distributed IDS by integrating the behavior based and knowledge based detection schemes. Previous efforts also include techniques such as game theoretical methods, visualization, and forensic traceback.[22–24]

To provide network security situation awareness,[25] in this paper we demonstrate a prototypical system that consists of both the distributed passive and active network sensors, which are designed to efficiently process network traffic and generate alerts. To present the useful data and obtain meaningful information for security analysts,[26,27] we develop several visualization features, including 1D, 2D and 3D (geolocation-based and Google map-based) traffic displays. To effectively detect attacks, we implement algorithms to transform the real-world information in IP addresses into images and investigate the patterns of threats. We also implement both the discrete wavelet transform (DWT) based approach and the traffic

For further author information, please send correspondence to Wei Yu: wyu@towson.edu and Genshe Chen:gchen@intfusiontech.com

Sensors and Systems for Space Applications VI, edited by Khanh D. Pham, Joseph L. Cox, Richard T. Howard,
Genshe Chen, Proc. of SPIE Vol. 8739, 87390R · © 2013 SPIE · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2015887

volume based approach to detect attacks. Through extensive simulation study, our data validates the effectiveness of our implemented system. For example, the DWT based approach obtains better performance than the traffic volume based approach.

The remainder of the paper is organized as follows: We introduce the design and implementation of our prototypical defense system in Section 2, including the testbed setup, attack traffic visualization, and attack detection. In Section 3, we show experimental results to validate the effectiveness of our implemented defense system. Finally, we give the conclusion in Section 4.

2. OUR PROTOTYPICAL SYSTEM

In this section, we first give the overview of our developed prototypical defense system. We then show the design and implementation of attack scene visualization, which can help human analysts to perform the cyber network situation awareness. After that, we show the implementation of the anomaly based detection using signal processing, image and statistical based schemes.

2.1 Overview

To detect cyber attacks, we develop a defense system that integrates the network detection information from different sources. Figure 1 depicts the baseline architecture of the system. In this defense system, both passive network sensors and active network sensors are deployed. The passive network sensors, which are deployed with IDS and distributed over the network, are used to monitor network traffic and identify suspicious attack behaviors. Note that the IDS based on anomaly-based detection commonly suffers from a high false positive rate because it tends to detect intrusions from a high noisy network environment. As a complimentary component, we consider leveraging the active network sensors. One example is to use the open source tool, Honeypots,[28, 29] which can be considered as a trap in the network to interact with the attacks and learn the insightful information of attacks. Because normal users do not actually use it, a low false positive rate can be achieved.

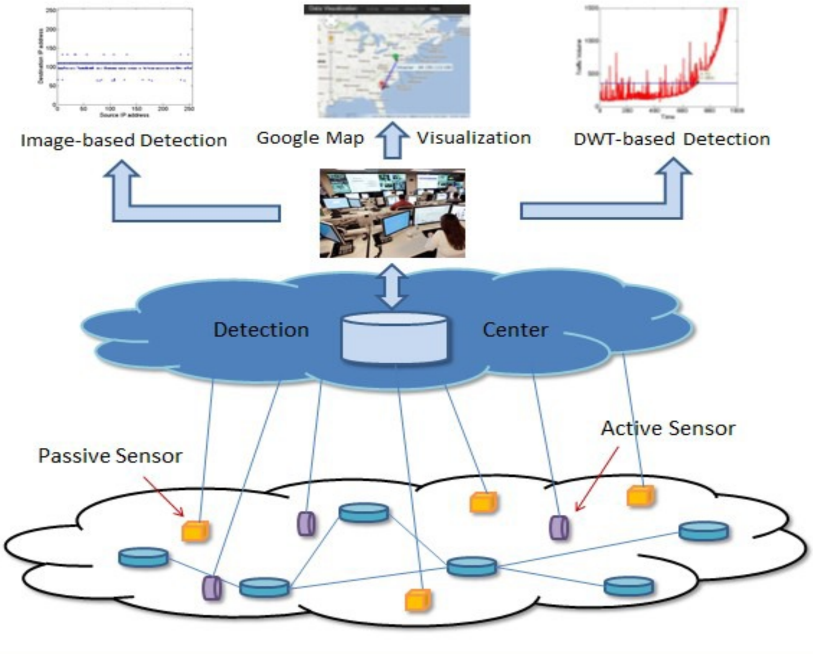


Figure 1: System Architecture

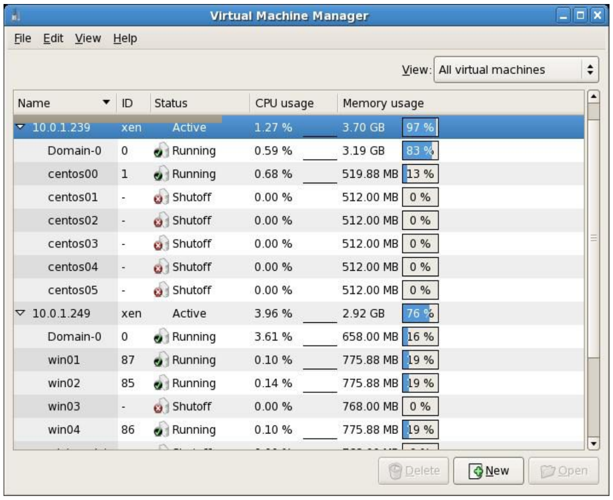
To monitor and control both passive and active sensors, the detection center is deployed to receive the detection logs and alerts from both the passive and active network sensors and stores the detection information into the database. Based on the collected information, we implement the anomaly based detection using digital signal processing and the image processing based schemes. In addition, to help the network administrator to conduct cyber security situation awareness, we design and implement a visualization component to display the geo-location and other information relevant to attacks on the layer of Google Map and Earth.[30]

2.2 Testbed Setup

To demonstrate the effectiveness of our proposed defense system, we implemented a prototypical system using both Windows Servers and Linux Servers to emulate an enterprise network. To show the exploit of security vulnerabilities and validate the effectiveness of detection schemes, we choose an unpatched Windows Server. We used BackTrack Linux, a Linux distribution with various hacking tools to emulate real-world cyber attacks.

As shown in Figure 3, our testbed consists of four computers (i.e., User, VMware host, XEN host dom0, and a detection center) and one router. On the XEN host machine dom0, we installed six virtual machines to emulate the enterprise network (virtual LAN 192.168.100.0/24), including servers and a HoneyD machine (as an active sensor). We installed the snort based IDS (as passive sensors) on dom0 so that it can monitor the traffic associated with the emulated enterprise network.

The attack network, located at the virtual LAN 192.168.200.0/24, is emulated on the VMware virtual machines hosted by the machine 192.168.1.22. Attacks are emulated using BackTrack. The detection center (192.168.1.25) runs a database server to store snort alert data from sensors. Based on the collected data, various anomaly detection and visualization algorithms can be provisioned. To enable the remote access for the system administrator, the visualization results can be viewed remotely from a password authorized laptop (192.168.1.10).



View: All virtual machines

Name	ID	Status	CPU usage	Memory usage	
▽ 10.0.1.239	xen	Active	1.27 %	3.70 GB	97 %
Domain-0	0	Running	0.59 %	3.19 GB	83 %
centos00	1	Running	0.68 %	519.88 MB	13 %
centos01	-	Shutoff	0.00 %	512.00 MB	0 %
centos02	-	Shutoff	0.00 %	512.00 MB	0 %
centos03	-	Shutoff	0.00 %	512.00 MB	0 %
centos04	-	Shutoff	0.00 %	512.00 MB	0 %
centos05	-	Shutoff	0.00 %	512.00 MB	0 %
▽ 10.0.1.249	xen	Active	3.96 %	2.92 GB	76 %
Domain-0	0	Running	3.61 %	658.00 MB	16 %
win01	87	Running	0.10 %	775.88 MB	19 %
win02	85	Running	0.14 %	775.88 MB	19 %
win03	-	Shutoff	0.00 %	768.00 MB	0 %
win04	86	Running	0.10 %	775.88 MB	19 %

Figure 2: Virtual Machine Manager

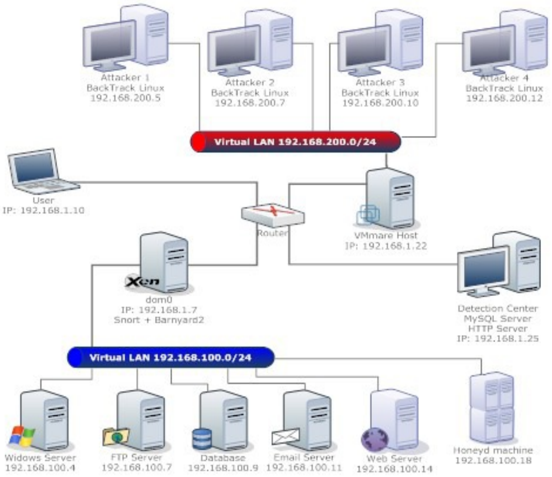


Figure 3: Testbed

Snort and Barnyard2 (BY2) are installed in the XEN host while the MySQL database and APACH-PHP-BASE (Basic Analysis and Security Engine) run in the detection center shown in Figure 3. Note that Snort is one of the widely deployed IDS systems. The most common alternative for handling output from Snort is to transmit data in a standard logging format (e.g., syslog) and store it into a database deployed via MySQL. Another alternative is to generate output in Snort's special unified format and processed through Barnyard2.

Active sensors are deployed by Honeyd and Arpd. Honeyd is a small daemon that creates virtual hosts on a network. The hosts can be configured to run various services. Arpd is a user-space ARP (Address Resolution Protocol) daemon. The ARP daemon moves the management of the ARP table from kernel to user space for the sake of efficiency. To use the ARP daemon, kernel needs to have ARPD and NETLINK support enabled. Arpd listens to ARP requests and answers for the IP addresses, which are unallocated. Using Arpd along with Honeyd, it is possible to populate the unallocated address space in the enterprise network with virtual Honeypots. The HoneyD and Arpd are installed in a virtual machine (HoneyD machine 192.168.100.18) so that the toolset can deploy Honeypots in the virtual LAN 192.168.100.0/24.

The main tasks of the detection center include maintaining MySQL based Snort log/alert database, provisioning anomaly detection algorithms, conducting the analysis of attack and defense interactions, visualizing attack patterns, and others. The MySQL server setup and Snort database configuration are shown as follows:

```
Step 1: Install MySQL server:
        yum install mysql-bench mysql-develphp-mysql
Step 2: Run and configure MySQL server:
        /usr/bin/mysql_secure_installation
Step 3: Create snort database:
        mysql -u root {p mysql>creat database snort; mysql>quit
Step 4: Create schema of snort database:
        wget http://www.snort.org/downloads/1631
        mysql -u root -p snort < schemas/create_mysql
Step 5: Authorize remote access from Xen host dom0 (192.168.1.7):
        mysql -u root -p
        mysql>grant create,insert,select,delete,update on snort.* to snort@"192.168.1.7"
        mysql>SET PASSWORD FOR snort@"192.168.1.7"=PASSWORD("iftift")
        use mysql
        select host,user,password from user
```

2.3 Attack Visualization

To build an effective defense system for conducting cyber security situation awareness, both automatic decision making and computer-aided data analysis are critical. In most cases, attack events can be accurately captured and processed by the detection algorithms, and the system can conduct corresponding actions such as adding suspicious entities into the blacklist. To provide meaningful information for human analysts, we develop a visualization component using PHP, HTML and JavaScript. PHP is used to fetch data and process data and processed data is then fed into the database through SYSLOG server. In the visualization component, we implement several visualization features such as 1D, 2D and 3D traffic displays. The 1D visualization is used to display the time-series of network traffic volume. The 2D visualization is used to display the correlation between source IP and destination IP addresses, which can be used to identify various attacks patterns. With the 3D visualization, the geological attributes of adversaries and victims can be displayed.

In terms of the 1D traffic visualization, the display of time sequence of network traffic volume is one basic way to visualize attacks. To this end, We implement the display of history traffic logs on an easy-to-use display interface. We also implement filters on the interface that enables the capability of filtering unrelated IP addresses. As the strength of suspicious traffic is an important indicator, we extract the traffic information from the alert data and visualize them in the form of bars. Through the length of bars associated with the IP addresses, we can provide the IP address information, which is most relevant to attacks.

In the 2D traffic visualization, we extract both the source and destination IP addresses from attack traffic log. The display on the 2D display is dynamically updated over time. As the output of our implemented detection schemes, we obtain a number of IP addresses designated as either victims or attack sources. With addition information, we can map the IP addresses to the physical locations (geolocation), which is defined as the identification of the real-world geographic location of an object. We develop one mapping scheme that takes the IP address as the input and returns the geolocation of the associated IP address. As a preliminary study, the mapping of IP addresses is at the level of city or town in our implemented system. More accurate information will be added in the future.

2.4 Attack Detection

We design and implement the anomaly detection against cyber attacks. We investigate the anomaly detection using the image based scheme to accurately detect anomalies in network traffic. Specifically, the image based detection (or textual recognition) schemes has the following stages. First, we use the packet headers of network traffic reported by passive network sensor in regular intervals and analyze the aggregated information at the detection center. Second, using the traffic data, we construct 2D images and use them to capture features and detect anomalies in network traffic. For example, we implement algorithms to detect patterns such as line or edge in 2D image and conduct the motion prediction to predict attack patterns. We implement a proof-of-concept image based detection against threats, including the worm/malware propagation denoted as many-to-many attacks, DDoS denoted as many-to-one attacks, and port-scanning attacks denoted as one-to-many attack. Based on the image based transform algorithm, we aggregate the collected traffic data into images, which facilitate the detection. With the image-based track algorithm, we can obtain IP addresses as attack sources or victims. The procedures of the image based transform and image based tracking are described in Algorithms 1 and 2, respectively.

In Algorithm 1, we first input i pairs of 4 digit source and destination IP addresses and define an array X called the traffic volume vector that stores the relevant pairs of IP addresses. Based on the collected data in this vector, we can easily display the relevant pairs of source IP addresses and destination IP addresses. In Algorithm 2, we initially load the color images generated by Algorithm 1 and then define an array G for keeping the gray-level value of each pixel that is transformed through conducting RGB-to-Gray transformation. After that, we derive the mean value and the standard derivation of array G and determine a threshold $m + kv$ to detect the anomaly where the sensitivity to detect attack can be adjusted by k. All the gray-level values beyond the threshold are stored and treated as the candidates of anomaly IP addresses.

Algorithm 1: The Image Based Transform Algorithm

```
Input    : Source IP Address $S_{i1}, S_{i2}, S_{i3}, S_{i4}$;
           Destination IP Address $D_{i1}, D_{i2}, D_{i3}, D_{i4}$;
           Traffic Volume Vector $X = [X_1, X_2, \ldots, X_8]$;
           Total Traffic Volume $M$;
Output   : Plot Image Based on Collected Data
Collect Traffic Data:
for $i = 1 : M$ do
    for $j = 1 : 4$ do
        $X_{ij} = S_{ij}$
        $X_{i(j+4)} = D_{ij}$          % Record Source IP and Destination IP address into Traffic Volume Vector
    end
end
Display Image of Traffic:
Input Traffic Volume Vector: $X = [X_1, X_2, \cdots, X_8]$;
for $i = 1 : M$ do
    for $j = 1 : 4$ do
        plot($X_{ij}, X_{i(j+4)}$)      % Plot Image Based on Collected Data
    end
end
Display 2D Image for Network Traffic
```

Algorithm 2: The Image Based Track Algorithm

```
Input    : Image of Traffic Data;
           Traffic Volume Vector: $G = [G_1, G_2, \cdots, G_8]$;
           Total Traffic Volume $M$;
           Threshold Parameter: $k$;
Output   : Print Attack IP Addresses
Gray Scale Image Generation:
$I = imread('image.jpg')$;        % Read Generated Color Image and save it in an Array
G= rgb2gray(I);                   % Transform Color Image into Gray Scale Image along with Gray-Level Value
Display Image of Network Traffic:
Input the Array of Gray-Level Value: $G = [G_1, G_2, \cdots, G_N]$;
Define $N$ as the Total Number of Array
$m = mean(G)$
$v = std(G)$                       % Calculate Mean Value and Stand Derivation of Data Stored in Array
for $i = 1 : N$ do
    if $G_i > m + kv$ then
        $j = i$;
    end
end
```

To conduct anomaly detection, we implement two detection schemes. Our first detection scheme uses traffic volume as the detection feature to detect attacks. It works in the following way: we first compute the mean and the standard deviation of traffic volume, where the mean m_1 and the standard deviation v_1 are the statistical measures of normal network traffic volume. We then determine a threshold as $h_1 = m_1 + k * v_1$, where k determines the degree of deviation from the original traffic. Using the real-world traffic data, we simulate the traffic volume based detection using MATLAB. The evaluation results are shown in Figure 4. As we can see, the original background traffic is random, and the blue line is the detection threshold. Note that such a threshold leads to some false positives. All traffic volumes below the value of blue line are normal traffic and the rest are traffic leading to false positives. We can see there is an obvious change in Figure 5 after we mix the attack and the background traffic while the threshold stays the same value. With the detection described above, we can find a detection time (e.g., $X : 635, Y : 220.9$ in Figure 5), indicating that all the traffic volumes larger than the detection threshold are marked as attack traffic.

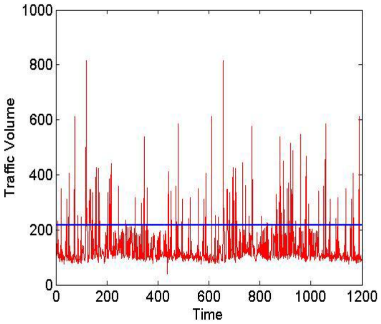


Figure 4: Background traffic

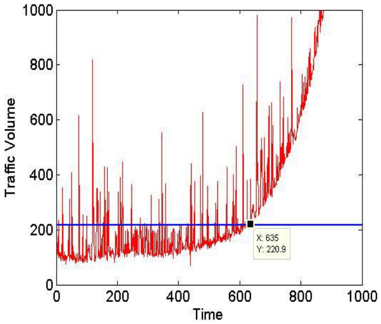


Figure 5: Mixture traffic with attack

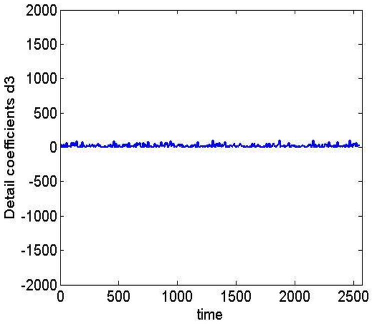


Figure 6: Background traffic

We also implement the discrete wavelet transform (DWT) based detection scheme and compare its effectiveness with the statistical based detection scheme. In particular, we load the original traffic data and conduct four-layer wavelet decompositions and reconstruct the multi-layer coefficients and collect the coefficients of the fourth layer. In Figure 6, we show the trend of coefficients in the original background traffic, which almost stays in the same level. In Figure 13, we show the change of coefficients in the mixture traffic with both attack and background traffic. From Figure 13, we see there is a sudden change after point $(621, 53.9)$, indicating that the traffic looks anomaly.

3. PERFORMANCE EVALUATION

In this section, we demonstrate the effectiveness of our developed defense system. We simulate several real-world attack scenarios and visualize attacks on Google Maps. The first scenario is *one-to-one* attack, where one adversary hacks one victim. In this attack, the adversary knows the specific victim and launches attack. In our traffic visualization, traffic between attack source and victim will be displayed. The second scenario is *one-to-many* attacks, in which the adversary chooses a number of different hosts to identify the vulnerable information on those hosts. One example of such attack is the port-scanning attack. The third scenario is a *many-to-many* attack that represents multiple adversaries attacking a number of victims.

Figure 7 shows a *one-to-one* attack scenario. As we can see, the green marker stands for the adversary while the red marker stands for the victim. The blue line means that the level of the alert is low. In this example, the adversary from one location in Maryland is attacking a victim in one place in Southern California. Moving the mouse on the marker can display a pop-up information window that shows the adversary and its IP address. Figure 8 shows a *one-to-many* attack scenario, in which one adversary in Maryland is attacking six victims in several states with various attack severity levels. Figure 9 illustrates a *many-to-many* attack scenario. In this example, four adversaries from Maryland, Ohio, Arkansas and Texas attacking twelve victims in several states with various severity levels are displayed.

Figure 10 shows a 3D display for *one-to-one* attack scenario. As we can see, the PlaceMaker with yellow color stands for the adversary while the red PlcaeMarker stands for the victim. The yellow line between them means the level of the alert is low. In this scenario, the adversary from one location in Kalispell is attacking a victim in a location in West Vancouver. Clicking the PlaceMarker will display a pop-up information window which shows the adversary and its Geolocation.

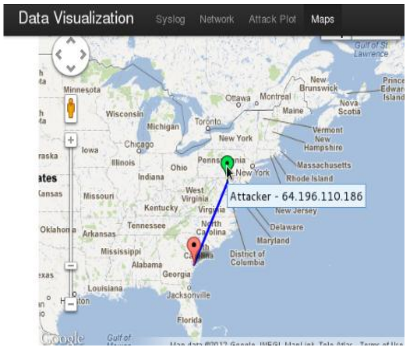


Figure 7: Geolocation of one-to-one attack

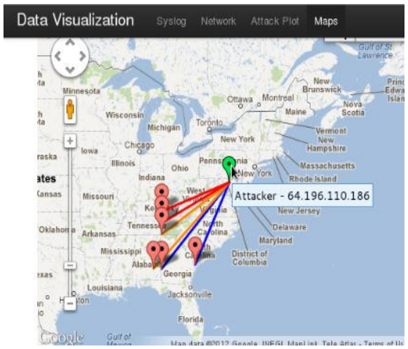


Figure 8: Geolocation of one-to-many attack

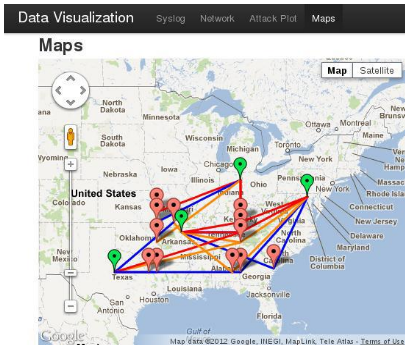


Figure 9: Geolocation of many-to-many attack

Figure 11 shows a *one-to-many* attack scenario where one adversary at Kalispell is attacking six victims in different cities. The attack to one victim in the location at Victoria and one victim in the location at Tacoma belong to the mid-level attack. The attacks to the victim in Vancouver belongs to the low severity attack, and the attack to hosts in Kentucky belongs to the high severity attack. Figure 12 illustrates a *many-to-many* attack scenario. Here we show that two adversaries from Kalispell and Calgary are attacking four victims in different cities with various severity levels.

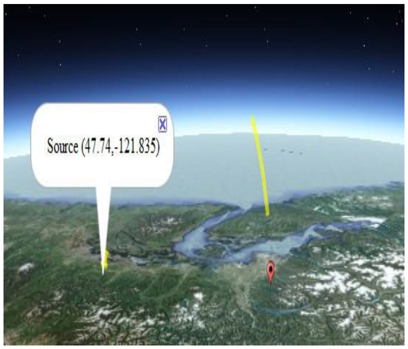


Figure 10: One-to-one attack

Figure 11: One-to-many attack

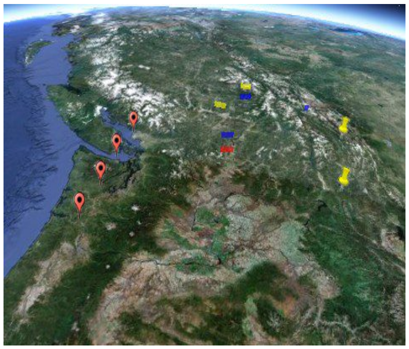


Figure 12: Many-to-many attack

We collect the real-world network traffic from a live system and background traffic. We then apply the malware propagation attack model to generate attack traffic. We use the following metrics to evaluate the effectiveness of implemented schemes: (i) Detection Rate (P_D), (ii) False Positive Rate (P_F), and (iii) Detection Time. Detection rate is defined as the probability of correctly determining the attack traffic if the attack occurs. False positive rate is the probability that a normal traffic is misclassified as attack traffic. Detection time is defined as time when the attack is accurately detected.

We simulate three typical attacks and validate the image based detection scheme. The attacks include DDoS attack (many-to-one), port-scanning (one-to-many), and malware propagation attack (many-to-many). Using DDoS as an example, the adversary uses a large number of IP addresses to attack a few destinations. We implement the image based transform algorithm and generated the images shown in Figure 14. The horizontal axis represents the source IP address and vertical axis represents the destination IP address. We can identify two possible attacks that are identified through the two horizontal lines.

To validate the effectiveness of DWT based detection scheme in comparison with the statistical based detection scheme, we conduct the simulation using real-world traffic traces. We use the following parameters: the amplitude of attack signal denoted as A and the frequency of attack signal denotes as f. To show tradeoff between P_D and P_f (commonly called a receiver operating characteristic (ROC) curve), we plot the relationship between P_D and P_f. Figure 15 shows the effectiveness of two detection schemes. As we can see, the DWT based detection scheme obtains better performance than the statistical based detection scheme. To compare the detection time of both detection schemes, we compare the detection

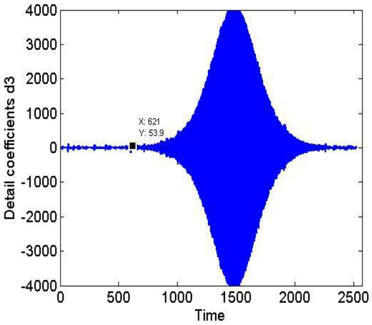


Figure 13: Mixture traffic with attack

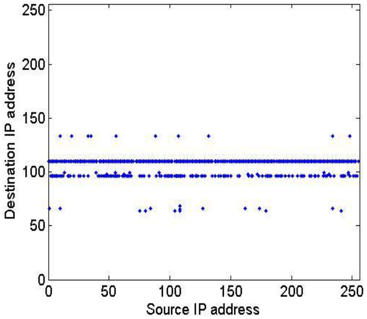


Figure 14: Image generated by related source and destination

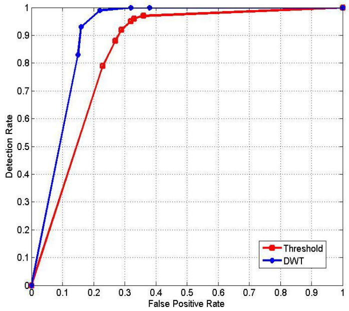


Figure 15: ROC of traffic volume Based and DWT Based detection

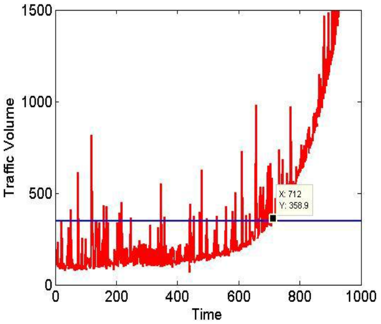


Figure 16: Detection Time of Traffic Volume Based Detection

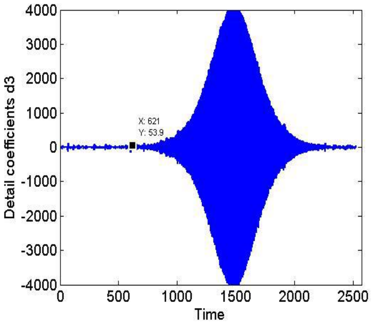


Figure 17: Detection Time of DWT Based Detection

rate given the same detection accuracy. For the statistical based detection scheme, we change the threshold and make it have the same detection accuracy as the DWT based detection. Figure 16 and Figure 17 show the detection time of both detection schemes. As we can see, the detection time of the statistical based detection scheme on traffic volume is 712 seconds and the DWT based detection scheme is 621 seconds, respectively. Hence, we conclude that the DWT based detection scheme can detect the attack more rapidly than the statistical based detection scheme.

4. FINAL REMARKS

In this paper, we implemented a prototypical defense system with both distributed passive and active network sensors. To present the useful cyber situational awareness information for human analysts, we developed several visualization features such as 1D, 2D and 3D traffic displays. To effectively detect attacks, we implemented algorithms to transform real-world traffic data into images and learn the pattern of attacks. We also leveraged both the DWT based scheme and the statistical based scheme to detect attacks. Through extensive experiments, our data validated the effectiveness of our implemented defense system.

REFERENCES

[1] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants," in *ACM SIGCOMM Conference*, pp. 75–86, 2003.

[2] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A Survey of Botnet Technology and Defenses," in *Cybersecurity Applications & Technology Conference For Homeland Security*, 2009.

[3] F. Yu, Y. Xie, and Q. Ke, "SBotMiner: large scale search bot detection," in *Web Search and Data Mining*, pp. 421–430, 2010.

[4] F. Giroire, J. Chandrashekar, N. Taft, E. M. Schooler, and D. Papagiannaki, *Exploiting Temporal Persistence to Detect Covert Botnet Channels*, 2009.

[5] S. Landau, M. R. Stytz, C. E. Landwehr, and F. B. Schneider, “Overview of Cyber Security: A Crisis of Prioritization,” *IEEE Security & Privacy* **3**, pp. 9–11, 2005.

[6] *I2P anonymous network*, `http://www.i2p2.de/`, 2010.

[7] G. Chen, D. Shen, C. Kwan, J. Cruz, M. Kruger, and E. Blasch, “Game theoretic approach to threat prediction and situation awareness,” *Journal of Advances in Information Fusion* **2**(1), pp. 1–14, 2007.

[8] D. Shen, G. Chen, E. Blasch, and G. Tadda, “Adaptive markov game theoretic data fusion approach for cyber network defense,” in *Proc. of IEEE MILCOM*, 2007.

[9] D. Shen, G. Chen, L. Hayes, and E. Blasch, “Strategies comparison for game theoretic cyber situational awareness and threat prediction,” in *Proceedings of 10th International Conference on Information fusion*, 2007.

[10] D. Shen, G. Chen, J. B. J. Cruz, M. Kruger, , and E. Blasch, “Game theoretic solutions to cyber attack and network defense problems,” in *Proceedings of 12th ICCRTs*, 2007.

[11] H. Chen, G. Chen, E. Blasch, M. Kruger, and I. Sityar, “Analysis and visualization of large complex attack graphs for networks security,” in *Proc. of SPIE, Vol. 6570*, 2007.

[12] M. Endsley, “Toward a theory of situation awareness in dynamic systems,” 1995.

[13] G. P. Tadda and J. S. Salerno, *Overview of Cyber Situation Awareness*.

[14] E. Blasch, J. J. Salerno, and G. Tadda, “Measuring the worthiness of situation assessment,” in *Proc. IEEE Nat. Aerospace Electronics Conf (NAECON)*, 2011.

[15] J. Salerno, E. Blasch, M. Hinman, and D. Boulware, “Evaluating algorithmic techniques in supporting situation awareness,” in *Proc. of SPIE, Vol. 5813*, 2005.

[16] D. Wagner and P. Soto, “Mimicry attacks on host-based intrusion detection systems,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 255–264, ACM, 2002.

[17] L. Li, G. Zhang, J. Nie, Y. Niu, and A. Yao, “The application of genetic algorithm to intrusion detection in mp2p network,” in *Advances in Swarm Intelligence*, pp. 390–397, Springer, 2012.

[18] K. Liang, H. Hon, M. Khairunnisa, T. Choong, and H. Khairil, “Real time intrusion detection system for outdoor environment,” in *Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on*, pp. 147–152, IEEE, 2012.

[19] D. Krishnan and M. Chatterjee, “An adaptive distributed intrusion detection system for cloud computing framework,” in *Recent Trends in Computer Networks and Distributed Systems Security*, pp. 466–473, Springer, 2012.

[20] H. F. Eid, A. Darwish, A. E. Hassanien, and T.-h. Kim, “Intelligent hybrid anomaly network intrusion detection system,” in *Communication and Networking*, pp. 209–218, Springer, 2012.

[21] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, “An efficient intrusion detection system based on support vector machines and gradually feature removal method,” *Expert Systems with Applications* **39**(1), pp. 424–430, 2012.

[22] D. Shen, G. Chen, L. Haynes, J. B. J. Cruz, M. Kruger, and E. Blasch, “A markov game approach to cyber security,” in *SPIE Newsroom, http://spie.org/x15400.xml?highlight=x2412*, 2007.

[23] G. Chen, H. Chen, E. Blasch, M. Kruger, and J. B. J. Cruz, “Information fusion and visualization of cyber-attack graph,” in *SPIE Newsroom, http://spie.org/x14562.xml*, 2007.

[24] W. Yu, X. Fu, E. Blasch, K. Pham, D. Shen, G. Chen, and C. Lu, “On effectiveness of hopping-based techniques for network forensic traceback,” in *Proc. of IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2013.

[25] E. Blasch, I. Kadar, J. Salerno, M. M. Kokar, S. Das, G. M. Powell, D. D. Corkill, and E. H. Ruspini, “Issues and challenges in situation assessment (level 2 fusion),” *Journal of Advances in Information Fusion* **1**(2), pp. 122–139, 2006.

[26] E. Blasch, “Sensor, user, mission (sum) resource management and their interaction with level 2/3 fusion,” in *Proc. of Int. Conf. on Info Fusion*, 2006.

[27] E. Blasch, “User refinement in information fusion,” in *Chapter 19 in Handbook of Multisensor Data Fusion 2nd Ed, Eds. M. E. Liggins, D. Hall, and J. Llinas, CRC Press*, 2008.

[28] N. Provos, “A Virtual Honeypot Framework,” in *USENIX Security Symposium*, pp. 1–14, 2004.

[29] H. Project, *Know Your Enemy: Learning About Security Threats*, Addison-Wesley, 2004.

[30] G. Chen, E. Blasch, D. Shen, H. Chen, and K. Pham, “Services oriented architecture (soa) based persistent isr simulation system,” in *Proc. of SPIE, Vol. 7694*, 2010.